Analysis and application of polling models

著者	Takagi Hideaki
発行年	1998
URL	http://hdl.handle.net/2241/530

No. 805

Analysis and Application of Polling Models

by

Hideaki Takagi

December 1998

Analysis and Application of Polling Models

Hideaki Takagi
Institute of Policy and Planning Sciences, University of Tsukuba
1-1-1 Tennoudai, Tsukuba-shi, Ibaraki 305-8573, Japan
e-mail: takagi@shako.sk.tsukuba.ac.jp

Polling models generally refer to systems of multiple queues served by nondedicated servers with rules that allocate the servers to the queues. The basic model consists of separate queues with independent Poisson arrivals served by a single server in cyclic order. Significant progress in the analysis of the basic and extended models, which began over forty years ago, has contributed to system performance evaluation techniques in such fields as computers, communications, manufacturing, and transportation, and has enriched queueing theory as well. This essay begins with the origin of the polling model, describes the basic polling model with its performance measures, and mentions three successful applications to the performance evaluation of communication networks. Several recent survey papers are referred to for further study.

1. Origin and Short History

Let me begin this essay by quoting a few passages from a nontechnical article by Dr. Martin A. Leibowitz (1968) entitled "Queues" in an old issue of *Scientific American* that refer to a polling model [11].

> Queuing theory was founded by the work of A. K. Erlang, who began in 1908 to study problems of congestion in telephone service for the Copenhagen Telephone Company. Since then workers in the field have tended to concentrate on describing fairly uncomplicated situations, involving at most a few queues. It seems likely that the pressures of a rapidly advancing technology will direct the attention of queuing theorists increasingly toward the analysis of systems containing many interacting queues. Such systems include the national and international telephone networks; large computers dealing with a variety of users or problems, and traffic-control systems covering wide areas.
>
> ... I should like to discuss what is called in queuing theory "the polling problem." Consider a large number of queues, each with its own input, served cyclically by a single server. The server goes from one queue to the next, taking a certain amount of travel time in the process, and serves everyone in a queue before proceeding to the next queue. A commonplace example would be a bus that stops at various places to load and unload passengers while it plies a circular route.
>
> My interest was stimulated by a different application, in which the server is a computer. It cyclically polls a number of remote terminals to ascertain

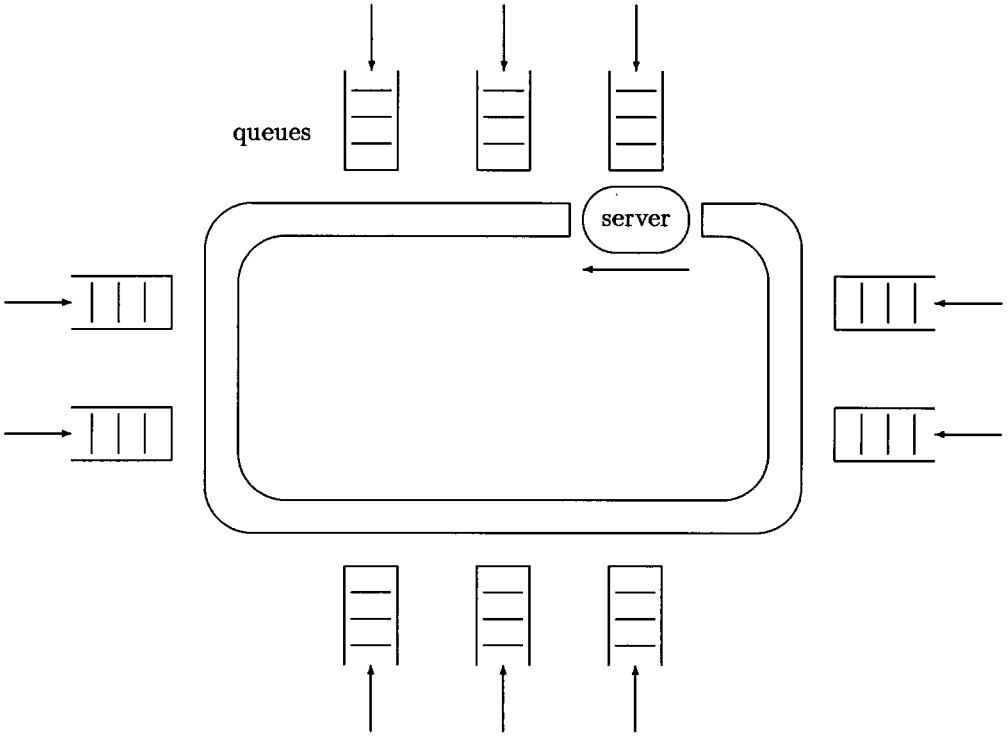


Fig. 1. A polling system.

> what demands have arrived since its last visit. This is the mode of operation in many time-shared systems.

As clearly described in the above quotation, a *polling model* in its most naive form is a system of multiple queues attended by a single server in cyclic order (Fig. 1). The term *polling* originated with the *polling* data link control scheme in which the central computer interrogates each terminal on a multidrop communication line to determine whether or not it has data to transmit. The addressed terminal transmits data, and the computer then examines the next terminal. Here, the server represents the computer and a queue corresponds to a terminal. This was an application of a polling model studied in the early 1970s. Situations represented by polling models and their variations appear not only in computers and communications but also in other fields of engineering such as manufacturing and transportation systems. The ubiquitous application is not surprising because the cyclic allocation of the server (resource) is a natural and simple way for fair arbitration to multiple queues (requesters). Therefore, polling models in various settings have been studied by many researchers since the late 1950s, focusing on the applications to technologies emerging at each period.

In the late 1950s, a polling model with a single buffer for each queue was used in an investigation of a problem in the British cotton industry involving a patrolling

machine repairman. In the 1960s, polling models with two queues were considered for the analysis of vehicle-actuated road traffic signal control. There are some early studies by queueing theoreticians, apparently independent of traffic analysis. In the 1970s, with the advent of computer communication networks, an extensive study was carried out on a polling scheme for data transfer from terminals on multidrop lines to a central computer. Around the early 1980s, the same model was revived for token passing protocols (e.g., token ring and token bus) in local area networks (LANs). More recently, polling models with additional control features (priority, time limit, etc.) have been applied to channel access protocols in metropolitan area networks (MANs), high-speed LANs, ISDN, and land mobile and satellite radio communication networks. In the application to computer systems, the polling model was used for the scheduling of moving arms in secondary storage devices, and for load sharing in multiprocessor computers. Numerous applications exist in manufacturing systems, such as assembly work on a carousel, an automated guided vehicle (AGV) system, and multiproduct economic lot scheduling. Applications to transportation systems include the public bus service on a circular route, an elevator on an up-and-down route, internal mail delivery, and shipyard loading for multiple destinations.

Regarding the mathematical analysis of polling systems, Dr. Leibowitz stated that "the complexity of such systems defies exact mathematical analysis," and proposed approximate modeling. Fortunately, however, subsequent progress in queueing theory has made it possible to handle the polling model in a much more tractable way than he envisaged somewhat pessimistically.

In this short essay, I would like to present an introductory overview of the analysis results of the polling model and its applications to the performance evaluation of several communication protocols. Besides hundreds of original research papers, polling models have been referred to in many books and survey papers on general communication networks as well as described in several dedicated surveys. Thus I will also include a brief discusion of these surveys and books for the convenience of those readers who want to proceed with further study. I conclude by suggesting some possible future directions.

2. Models and Performance Measures

This section presents the polling model and its analysis results. Let me mainly describe the very basic system for the sake of conciseness, and refer to its variations only briefly. My description shall be given in terms of queueing theory, with which the reader is assumed to have some familiarity. The generic queueing theoretic terms should correspond to the specific technical terminologies in each application context.

A *basic polling system* consists of several, say N, queues served by a single server in cyclic order. Assume that the characteristics of all queues are identical. Customers arrive at each queue according to an independent Poisson process at rate λ. Let b and $b^{(2)}$ be the mean and the second moment, respectively, of the service time. The total load offered to the system is then given by

$$\rho = N\lambda b.$$

The server walks through all the queues in cyclic order. Let r and δ^2 be the mean and the variance, respectively, for the time needed by the server to switch from one queue to the next. These switchover times are assumed to be independent of the arrival and service processes. The total mean switchover time per polling cycle is then given by

$$R = Nr.$$

Two important performance measures of the polling system are the *mean polling cycle time* C that it takes the server to complete a cycle of visiting all the queues in the system, and the *mean customer waiting time* W that it takes a customer from arrival to service start. Analytical results are available in closed form for C and W in the two extreme cases, i.e., a single-buffer system and an infinite-buffer system when all the queues are statistically identical.

2.1. Single-buffer systems

A system in which each queue can accommodate at most one customer at a time is called a *single-buffer system.* Those customers that arrive to find the buffer occupied are lost. This system can model a *patrolling machine repairman problem* (in which the breakdown of a machine corresponds to the customer arrival and the patrolling repairman to the roving server) and an interactive transaction processing system on a computer shared by multiple users. If both service times and switchover times are constant ($b^{(2)} = b^2, \delta^2 = 0$) in a single-buffer system, the mean cycle time and the mean waiting time are given by

$$C = R + Qb$$

and

$$W = (N-1)b - \frac{1}{\lambda} + \frac{NR}{Q},$$

where

$$Q = \frac{N \sum_{n=0}^{N-1} \binom{N-1}{n} \prod_{j=0}^{n} \left[e^{\lambda(R+jb)} - 1\right]}{1 + \sum_{n=1}^{N} \binom{N}{n} \prod_{j=0}^{n-1} \left[e^{\lambda(R+jb)} - 1\right]}$$

is the mean number of customers served in a polling cycle. The *throughput* γ of the system, or the mean number of customers served per unit time, is given by

$$\gamma = \frac{Q}{C} = \frac{N}{W + b + 1/\lambda}.$$

Fig. 2 plots W against ρ when $b = R = 1$ for $N = 5$, 20, 100, and ∞. For N finite, $W \approx R + (N-1)b$ when $\rho \to \infty$ as all the queues are occupied most of the time. The throughput then approaches its maximum, the *capacity* of the system $\gamma_{\max} = N/(R + Nb)$. On the other hand, $W \approx R/2$ when $\rho \to 0$.

Taking the limit $N \to \infty$ with ρ and R fixed at finite values, one obtains the *continuous polling model* in which the server travels at constant speed around a circular

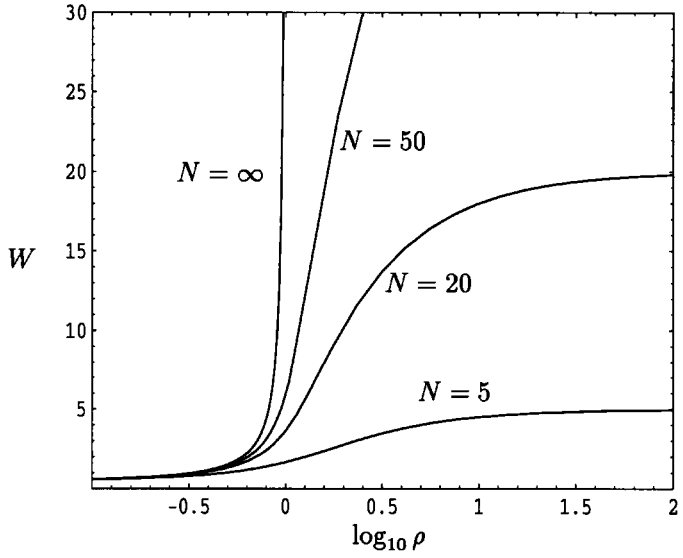


Fig. 2. Mean waiting time in a single-buffer polling system ($b = R = 1$).

route on which customers arrive at points uniformly distributed over the circle. In this case we get

$$C = \frac{R}{1-\rho} \quad ; \quad W = \frac{R + \rho b}{2(1-\rho)}.$$

2.2. Infinite-buffer systems

At the other extreme, a system in which any number of customers can wait without loss upon arrival at each queue is called an *infinite-buffer system.* It is natural to assume that if the server finds at least one customer at a queue it visits, service is immediately started there. However, several rules can be considered with respect to the instant at which the server leaves the queue. The following three rules are typical among others. In an *exhaustive service* system, the server continues to serve each queue until it empties. Customers that arrive at the queue being served are also served in the current service period. In a *gated service* system, the server serves only those customers that were found waiting when it visited the queue. Those that arrive at the queue during its service period are set aside to be served in the next round of polling. In a *limited service* system, at most one customer is served at each visit to a queue.

For a broad class of infinite-buffer systems, including those with the three typical service rules mentioned above, the mean cycle time is simply given by

$$C = \frac{R}{1-\rho}$$

if $\rho < 1$, which is the condition for the stability of the system. The mean waiting times

ρ	C	W		
		exhaustive	gated	limited
.00	1.0000	.5500	.5500	.5500
.05	1.0526	.6000	.6053	.6085
.10	1.1111	.6556	.6667	.6742
.15	1.1765	.7176	.7353	.7485
.20	1.2500	.7857	.8125	.8333
.25	1.3333	.8667	.9000	.9310
.30	1.4286	.9571	1.000	1.045
.35	1.5385	1.062	1.115	1.179
.40	1.6667	1.183	1.250	1.339
.45	1.8182	1.327	1.409	1.535
.50	2.0000	1.500	1.600	1.778
.55	2.2222	1.711	1.833	2.089
.60	2.5000	1.975	2.125	2.500
.65	2.8571	2.314	2.500	3.070
.70	3.3333	2.767	3.000	3.913
.75	4.0000	3.400	3.700	5.286
.80	5.0000	4.350	4.750	7.917
.85	6.6667	5.933	6.500	15.00
.90	10.000	9.100	10.00	100.0
.95	20.000	18.60	20.50	-

Table 1. Mean cycle time and mean waiting time in an infinite-buffer polling system ($N = 10, r = 0.1, \delta^2 = 0.01, b = 1, b^{(2)} = 1$).

are given as follows:

$$W_{\text{exhaustive}} = \frac{\delta^2}{2r} + \frac{N\lambda b^{(2)} + r(N-\rho)}{2(1-\rho)},$$

$$W_{\text{gated}} = \frac{\delta^2}{2r} + \frac{N\lambda b^{(2)} + r(N+\rho)}{2(1-\rho)},$$

$$W_{\text{limited}} = \frac{\delta^2}{2r} + \frac{N\lambda b^{(2)} + r(N+\rho) + N\lambda\delta^2}{2(1-\rho-N\lambda r)}.$$

Table 1 shows the numerical values of the mean cycle time and the mean waiting times for the three rules. The mean waiting times are ordered as

$$W_{\text{exhaustive}} \leq W_{\text{gated}} \leq W_{\text{limited}}.$$

In fact it has been proven mathematically that the exhaustive service rule stochastically minimizes the unfinished work and the number of customers in the system at all times. However, an operational advantage of the limited service rule is the *fairness* in service

opportunity in the sense that it prevents heavily loaded queues from monopolizing the server.

2.3. Variations

Numerous variations and extensions to the above-mentioned basic systems have been proposed and analyzed. Some have been introduced to model the operation of practical systems more precisely, while others seem to have been made up as academic exercises. Let me focus on major variations briefly here. Readers who are interested in the numerous variations of the basic system and their detailed analysis may search the literature by first consulting the survey papers mentioned in Section 4.

(1) Asymmetric systems

Each queue may have different stochastic characteristics such as different arrival rate, service time distribution, and switchover time distribution. In this case, the mean waiting time can be obtained through the numerical solution to the set of linear equations for the single-buffer system and for the infinite-buffer systems with exhaustive and gated service rules.

A theoretically important and practically useful relationship valid for asymmetric infinite-buffer systems with mixed service rules is the *pseudoconservation law*

$$\sum_{i\in E,G} \rho_i W_i + \sum_{i\in L} \rho_i \left(1 - \frac{\lambda_i R}{1-\rho}\right) W_i = \frac{\sum_{i=1}^{N} \rho_i b_i^{(2)}}{2(1-\rho)} + \frac{\rho\Delta^2}{2R} + \frac{R\left(\rho - \sum_{i=1}^{N} \rho_i^2 + 2\sum_{i\in G,L} \rho_i^2\right)}{2(1-\rho)}$$

where $\rho_i = \lambda_i b_i$, and subscript i refers to the quantity associated with queue i. Δ^2 is the variance of the total switchover time. E, G, and L stand for the index sets of the queues with exhaustive, gated, and limited service rules, respectively. While it does not yield individual W_i's, the merits of the pseudoconservation law include a measure of overall performance, a demonstration of the effects of various parameters on W_i's, a basis for approximations and bounds, and a validity check of simulation results. When there are no switchover times, the pseudoconservation law reduces to *Kleinrock's conservation law* for nonpreemptive work-conserving queueing systems:

$$\sum_{i=1}^{N} \rho_i W_i = \frac{1}{2(1-\rho)} \sum_{i=1}^{N} \rho_i b_i^{(2)}.$$

(2) Discrete-time and/or batch arrival systems

A motivation for discrete-time system models comes from clock-synchronized operation in digital computers and communications, where time is slotted and the service times are multiples of the slot size. The batch arrival of customers models the transmission unit consisting of several messages, packets, frames, cells, etc. These features can be incorporated in the analysis of basic systems without many additional difficulties.

(3) Noncyclic polling order and optimization

There are many systems in which the server does not visit all the queues exactly in cyclic order. For example, the physical structure of the system may require the server to visit queues first in one direction and then in the reverse direction. Such cases apply to the elevator in a building and to the scanning policy in the moving-arm disk device of a computer. Systems may be designed so as to visit some queues more often than others in a cycle to establish priority service; the polling order table was implemented in the multidrop data link controller. A system with Markovian server movement (the server visits queue j after queue i with given probability p_{ij}) has also been proposed; a special case is random polling in which the server chooses the next queue to serve completely at random. For those systems in which the server movement does not depend on the system state such as the queue lengths, one can again extend the analysis of the basic system rather easily.

In other systems, the server's movement/action may be controlled based on the local or global system state. For example, service may not start unless a large enough number of customers are found upon a visit, as sometimes is seen in manufacturing and transportation systems. As another example, the server may visit the longest queue in the system after each service. If there are no customers in the system, what should the server do? Options include to keep cycling, staying at the last visited queue, going back to some home base queue, and waiting at the queue with the highest static load. Many optimization problems with respect to the server movement remain unsolved theoretically.

(4) Network of queues

In usual queueing networks, each queue has it own dedicated server(s). In a polling system for a network of queues, there is a single server visiting all the queues one at a time, while the customers served at a queue may move to other queues or depart from the network. Analytical work has been done for both open and closed networks, but useful applications seem to remain unidentified.

(5) K-limited and time-limited service

Under a limited service rule, at most one customer is served at each visit of the server to the queue. This has been extended to the *K-limited service* rule in which the service continues until K customers are served or the queue is empty, whichever occurs first. The pipeline polling scheme in satellite communication is such an example. Another extension is the *time-limited service* rule in which the duration of server attendance to each queue (instead of the number of customers served) is limited. This is important from the application viewpoint because several communication protocols implement timed-token operation. Unfortunately, the exact analysis of a polling system with time-limited service rule is very difficult, and the results do not seem to be readily usable for practitioners yet. Alternatively, simple approximations have been proposed.

3. Application to Communication Networks

In this section, I will highlight three successful applications of the polling model to the performance evaluation of communication networks. They are rather classical, but simple and therefore instructive. Again, those who want to know more about application examples are referred to the survey papers mentioned in Section 4.

3.1. Half duplex transmission

Half duplex transmission is a mode of transmitting data between two parties on a shared communication line. Transmission is possible in either direction but not in both directions simultaneously. (A similar situation can be observed in everyday life, e.g., a traffic light at the intersection of two one-way streets, a narrow bridge or passage, conversation with Walkie-Talkie.) Suppose that a central computer and a communication control unit (ccu) connected to several data terminals exchange messages over a half duplex line (Fig. 3). When the transmission from the ccu is complete, a finite time is needed to reverse the direction of transmission on the line. Output messages are sent from the computer to the ccu, which delivers them to the terminals. After receiving a polling message from the computer and again reversing the direction of transmission, the ccu can start sending input messages to the computer, and this cycle is repeated.

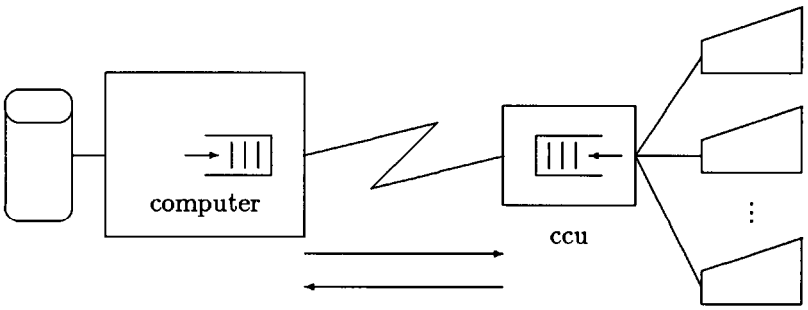


Fig. 3. Half-duplex transmission for an inquiry system.

In the queueing model of this system, customers correspond to the input and output messages, and the server represents the communication line between the computer and the ccu that allows the alternating transmission of messages. Let us call the computer queue 1 and the ccu queue 2. The service time at queue 1 is the transmission time of an output message, and that at queue 2 is the transmission time of an input message. The switchover time from queue 1 to queue 2 consists of the time for sending the polling message and the facility reversal time. The switchover time from queue 2 to queue 1 consists only of the facility reversal time.

Let us introduce parameters for the above quantities. Let λ_1 and λ_2 (messages/sec) be the arrival rates of the output and input messages, respectively. Let $\overline{X_1}$ and $\overline{X_1^2}$ be the mean and the second moment of the length (in characters) of an output message. Similarly, we use $\overline{X_2}$ and $\overline{X_2^2}$ as the mean and the second moment of the length of an input message. The transmission speed of the line is denoted by S (characters/sec).

The transmission time of a polling message is given by a constant P_1 (sec). Finally, the constant facility reversal time is denoted by t_r (sec). We can neglect the signal propagation time in this system, as the line speed is usually very low. These parameters of the system are converted into the parameters of an asymmetric polling model with two queues as follows. For queue i, $i = 1, 2$, b_i and $b_i^{(2)}$ are the mean and the second moment of the service time, respectively, and r_i is the (constant) switchover time from queue i to the other queue. We then have

$$b_i = \frac{\overline{X_i}}{S}, \quad b_i^{(2)} = \frac{\overline{X_i^2}}{S^2} \; (i = 1, 2) \quad ; \quad r_1 = P_1 + t_r \quad ; \quad r_2 = t_r.$$

Since our system is asymmetric, we cannot use the formula for symmetric queues in Section 2.2. Instead we use

$$W_1 = \frac{\lambda_1 b_1^{(2)}}{2(1-\rho_1)} + \frac{\lambda_1 \rho_2^2 b_1^{(2)} + \lambda_2 (1-\rho_1)^2 b_2^{(2)}}{2(1-\rho_1)(1-\rho)(1-\rho+2\rho_1\rho_2)} + \frac{(1-\rho_1)R}{2(1-\rho)}$$

and a similar formula for W_2 obtained by exchanging subscripts 1 and 2, where

$$\rho_i = \lambda_i b_i \; (i = 1, 2) \quad ; \quad \rho = \rho_1 + \rho_2 \quad ; \quad R = r_1 + r_2.$$

Sykes [16] applies this model to the calculation of the *mean retrieval time* T_r in an inquiry-response system, which is given by

$$T_r = W_2 + b_2 + C_p + W_1 + b_1,$$

where C_p is the mean time for processing an inquiry at the computer. He uses the values

$$S = 120 \text{ characters/sec} \quad ; \quad \overline{X_2} = 15 \text{ characters}$$
$$c_1^2 = \mathrm{Var}[X_1]/\overline{X_1}^2 = 0.1 \quad ; \quad c_2^2 = \mathrm{Var}[X_2]/\overline{X_2}^2 = 0.5$$
$$t_r = 0.2 \text{ sec} \quad ; \quad P_1 = 0.2 \text{ sec} \quad ; \quad C_p = 2.0 \text{ sec}$$

and assumes that $\lambda_1 = \lambda_2$, because each inquiry is assumed to generate a response. In Fig. 4, we plot the mean retrieval time T_r against the line utilization ρ for $\overline{X_1} = 75$, 150, and 300 characters.

3.2. Polling data link control

Polling control has often been employed in network configurations in which geographically dispersed terminals are connected to a central computer in a tree topology or a loop topology (Fig. 5). There are two types of polling control. In *roll-call polling*, the computer has a polling sequence table according to which it interrogates each terminal. The addressed terminal then transmits all waiting messages to the computer. When the transmission from one terminal is complete, the computer starts polling the next terminal. In the polling sequence table, the network designer can order terminals in exact cyclic order, or in any sequence and frequency to prioritize terminals. Roll-call polling is suitable for a tree topology. In a loop topology, on the other hand, *hub polling* is often used. In this case, a natural polling sequence is determined by the position

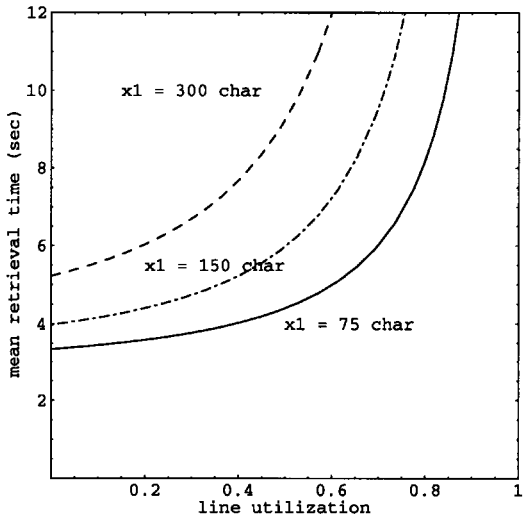
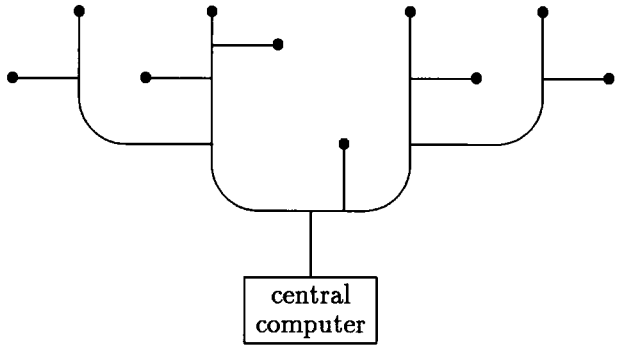


Fig. 4. Mean retrieval time in an inquiry-response system with a half-duplex transmission.

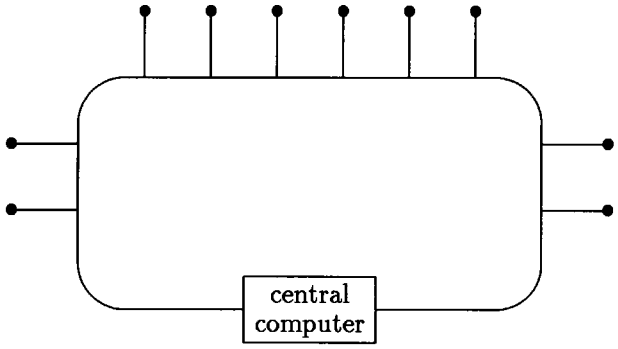
of terminals on the loop. The central computer initiates polling by interrogating the terminal at the end of the loop. This terminal transmits its waiting messages, to which it appends a polling message for the next downstream (in the direction of transmission) terminal. The latter terminal similarly adds its own messages followed by another polling message, and so on. At the completion of a polling cycle, the central computer collects all the messages and assumes control.

Polling was implemented in byte-oriented protocols, such as the early IBM Programmed Airlines Reservation System (PARS) and IBM's Binary Synchronous Communication (BSC). Polling has also been implemented in bit-oriented protocols such as IBM's Synchronous Data Link Control (SDLC), ANSI's Advanced Data Communication Control Procedure (ADCCP), and CCITT's High-level Data Link Control (HDLC). SDLC provides hub polling under the name *SDLC loop*.

The operation in normal response mode (NRM) of HDLC is as follows. A single station (usually the central computer) designated the *primary station* is responsible for control of the link, issuing commands to terminals called *secondary stations*, and receiving responses from them. The fifth bit of the control (C) field in the HDLC frame is the polling/final (P/F) bit. The primary station initiates a polling cycle by transmitting a receive-ready (RR) frame with P=1 to the first secondary station in the polling sequence table. The addressed secondary station transmits waiting messages in information (I) frames. A station can only transmit up to seven I frames per poll without waiting for an acknowledgment. If there are no errors and the primary station is ready to accept further frames, a positive acknowledgment is sent from the primary station. The secondary station starts to transmit messages again, and continues in this way until it transmits the last frame with F=1, indicating the completion of its reply



(a) tree topology.



(b) loop topology.

Fig. 5. Typical topologies for polling networks.

to this poll. The primary station acknowledges this transmission, and polls the next secondary station.

For simplicity of analysis, let us employ a symmetric model so that the formulas in Section 2.2 can be used. The assumptions for this model include the following. N terminals are connected to a computer. The arrival processes at each terminal are independent Poisson processes with equal arrival rates of λ messages/sec. The switchover time between adjacent terminals, which depends on the network topology and whether roll-call or hub polling is used, is the same for every adjacent pair, and is further assumed to be constant ($\delta^2 = 0$), so that the round-trip switchover time is R sec. The message length distributions are the same for each station; if $\overline{X}$ and $\overline{X^2}$ are the mean and the second moment of the message length in bits, and S (bits/sec) is the line speed, we have

$$b = \frac{\overline{X}}{S} \quad ; \quad b^{(2)} = \frac{\overline{X^2}}{S^2}.$$

Finally, the polled terminal continues to transmit queued messages until it empties (exhaustive service). Under these assumptions, we get

$$W = \frac{N\lambda b^{(2)} + R(1-\rho/N)}{2(1-\rho)}.$$

The switchover time R consists of the following components. In roll-call polling, it includes the time t_P taken to transmit a polling message to each terminal (if an RR frame of 48 bits long is used in HDLC, $t_P = 48/S$ sec), the synchronization time t_S for a terminal to recognize its address and take action to begin transmitting, and the propagation time required for polling messages and data messages to physically propagate on the transmission line. If $\tau_{\text{roll-call}}$ denotes the total propagation time for the entire system, the total switchover time for roll-call polling is given by

$$R_{\text{roll-call}} = Nt_P + Nt_S + \tau_{\text{roll-call}}.$$

Note that $\tau_{\text{roll-call}}$ depends on the topology and the line length of the network. The total switchover time is reduced through the use of hub polling (even for the same topology and line length) for two reasons. First, the propagation time is reduced, since there is no back-and-forth transmission of polling and response messages for each terminal. Thus, the propagation time is just one round-trip delay, denoted by τ_{hub}. Second, the polling message is transmitted by the computer only once to the first terminal, and then propagated downstream in cyclic polling. If we denote this contribution by t'_P, the total switchover time for hub polling is

$$R_{\text{hub}} = t'_P + Nt_S + \tau_{\text{hub}}.$$

Hence, hub polling usually has higher performance than roll-call polling, particularly for large-scale networks.

As a numerical example, Schwartz [14, sec. 8.1] considers the network topology shown in Fig. 6, where the length of the line from the central computer to the Nth terminal is denoted by l (miles). Let τ be the round-trip propagation time (milliseconds) between the computer and the Nth terminal. The signal propagation time is assumed to be 2 milliseconds per 100 miles. Thus

$$\tau = l \text{ (miles)} \times 2 \times \frac{2 \text{ milliseconds}}{100 \text{ miles}}.$$

For the topology of Fig. 6, we obviously have

$$\tau_{\text{roll-call}} = \frac{\tau}{2}(1+N) \quad ; \quad \tau_{\text{hub}} = \tau.$$

Schwartz uses the values

$$S = 4{,}800 \text{ (bits/sec)} \quad ; \quad N = 10 \quad ; \quad t_P = t'_P = t_S = 10 \text{ millisec}$$

$$\overline{X} = 1{,}200 \text{ bits} \quad ; \quad \overline{X^2} = 2\overline{X}^2 = 2{,}880{,}000 \text{ (bits)}^2 \quad \text{(exponential distribution)}.$$

Using these values and $l = 2{,}000$ miles or 100 miles, we plot the mean message waiting times W for both roll-call and hub polling in Fig. 7. This figure illustrates that the difference in the two polling schemes becomes evident for a long network.

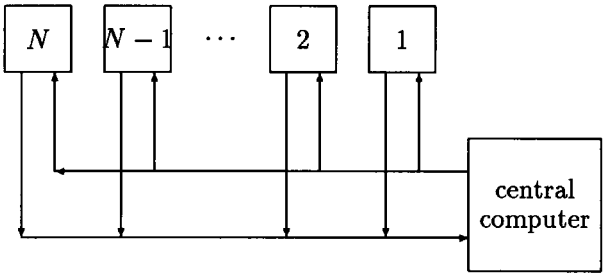


Fig. 6. Model of a polling network.

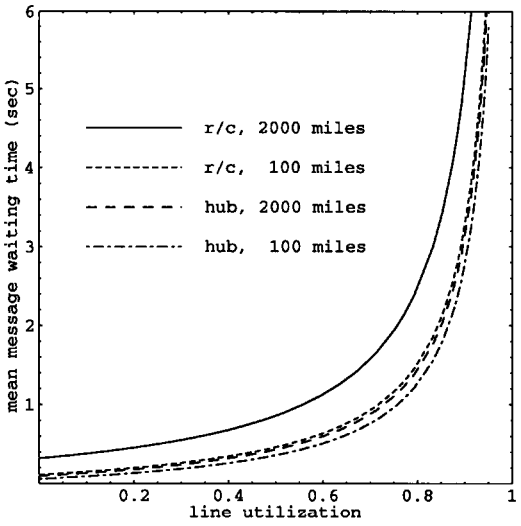


Fig. 7. Mean message waiting times under two polling schemes.

3.3. Token ring network

Two major types of access schemes for local area computer networks developed in the early 1980s were the *random access* scheme represented by Carrier Sense Multiple Access/Collision Detection (CSMA/CD) implemented in Xerox Ethernet, and the *controlled access* scheme represented by the *token ring* protocol developed at IBM Zürich Research Laboratory. They were later included in IEEE 802 standards as 802.3 for a CSMA/CD bus, 802.4 for a token-passing bus, and 802.5 for a token-passing ring. Token-ring and token-bus are examples of networks in which *explicit token* messages are used to control the transmission rights.

A ring network is a group of *stations* (terminals or computers) that are interconnected via a communication line in the form of a loop (Fig. 8). (The words *ring* and *loop* are used interchangeably). Each station is attached to the ring by an interface unit. Traffic on the communication line is usually unidirectional, so that each station receives messages from one of its neighbors and passes them to its other neighbor. Messages sent from a source station to a destination station are relayed by intermediate stations. A ring supervisor may be present (for initialization, access control, congestion moni-

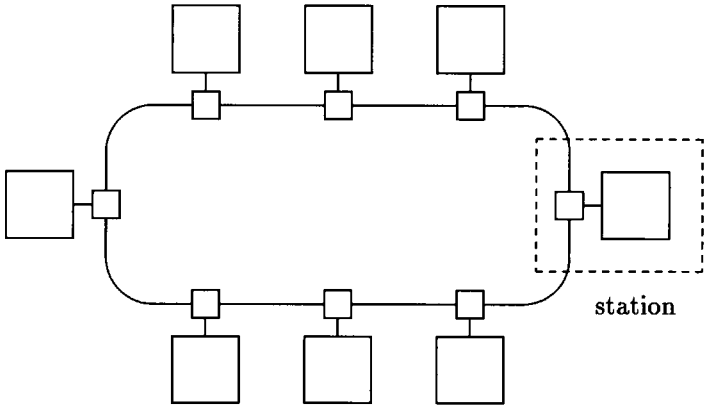


Fig. 8. Configuration of a ring network.

toring, statistics gathering, error recovery, etc.), or stations may control themselves in a distributed manner.

In a token ring network, a number of stations are connected to the ring network through an adapter, and all information goes through each station when it circulates. A permit to transmit is controlled through the use of a *token*, which is passed from station to station according to a given rule. In the IEEE 802.5 token ring, the fourth bit of the access control (AC) field in a frame specifies whether the token is free or busy. A station that receives a free token modifies the bit to busy, and inserts the destination address (DA), its own address (SA), and messages (INFO) when it sends out the frame. The transmitting station is responsible for removing its frame from the ring and for generating a new free token when its transmission is over.

With respect to the time at which a new free token is generated by the transmitting station, distinction is made for the *multiple-token*, *single-token*, and *single-message* operations. For multiple-token operation, the transmitting station generates a new free token and places it immediately after the last bit of a transmitted message. Therefore, for a long ring, the chances are that there are several message frames and a free token frame on the ring at one time. For single-token operation, a new free token is generated by erasing the busy token bit in the header of a transmitted message when it returns. If a message is long, the transmitting station will receive the busy token before it has finished transmitting. In this case, a free token is generated only after the last bit of a message has been transmitted, as in the multiple-token operation. For single-message operation, a new free token is generated only after the last bit of the transmitted message has been erased. Both single-token and single-message operations ensure that there is at most one free or busy token on the ring at all times.

Let us proceed to the performance modeling of a token ring network, following Hammond and O'Reilly [7, sec. 8.2]. We assume that N stations are connected to a ring, and that the arrival process at each station is a Poisson process with a rate of λ messages/sec. Let $\overline{X}$ and $\overline{X^2}$ be the mean and the second moment of the message

length in bits, and S the ring speed in bits/sec. Since bits are processed serially at each station interface, a delay called the *station latency* is needed for the output to be passed on to the ring. The station latency is at least one bit and can range up to a dozen bits, depending on the control feature; it is denoted by B bits. Finally, let τ (sec) be the propagation time for the ring. If l (kilometers) is the length of the ring, a good estimate of τ is given by

$$\tau = l \times 5 \times 10^{-6} \text{ sec}$$

on the assumption that the signal propagates at two-thirds of the speed of light. The time for the circulation of a token is then given by

$$R = \tau + \frac{NB}{S}.$$

As a performance measure, let us consider the mean time T_f from the instant at which a message arrives at a transmitting station to the instant that it is received at a destination station. If we assume that destination stations are uniformly distributed over the ring, the mean propagation time from a transmitting station to a destination station is given by $R/2$. Since there are no simple results available for a polling model with a time-limited service rule as in the ANSI/IEEE 802.5 Standard, we will instead use W for the exhaustive service polling system given in Section 2.2 to compute the mean time from the arrival to the transmission start of a message. Hence the mean message transfer time is given by

$$T_f = \frac{\overline{X}}{S} + \frac{R}{2} + \frac{N\lambda b^{(2)}}{2(1-\rho)} + \frac{R(1-\rho/N)}{2(1-\rho)},$$

where $\rho = N\lambda b$, and b and $b^{(2)}$ are determined according to the free-token generation policies discussed above. When this formula is applied to multiple-token operation we use

$$b = \frac{\overline{X}}{S} \quad ; \quad b^{(2)} = \frac{\overline{X^2}}{S^2}.$$

For single-token operation, we have

$$b = R + \frac{1}{S}\int_{SR}^{\infty}[1 - B(x)]dx \quad ; \quad b^{(2)} = R^2 + \frac{2}{S^2}\int_{SR}^{\infty} x[1 - B(x)]dx,$$

where $B(x)$ is the probability distribution function of the message length X. For single-message operation, we have

$$b = \frac{\overline{X}}{S} + R \quad ; \quad b^{(2)} = \frac{\overline{X^2}}{S^2} + \frac{2\overline{X}R}{S} + R^2.$$

Substituting these into T_f, we get

$$T_f(\text{single-message}) > T_f(\text{multiple-token}),$$
$$T_f(\text{single-message}) > T_f(\text{single-token}).$$

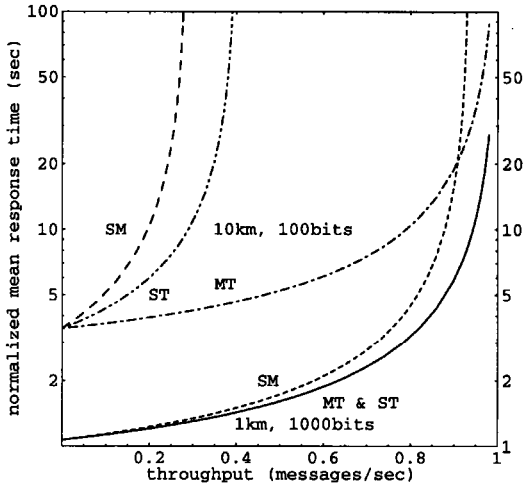


Fig. 9. Normalized mean transfer time in token ring networks.

For numerical comparison of the mean transfer times for the three operations, we assume a constant message length and consider the normalized mean transfer time given by

$$\frac{T_f}{\overline{X}/S} = 1 + \frac{a}{2} + \frac{\rho b}{2(1-\rho)(\overline{X}/S)} + \frac{a(1-\rho/N)}{2(1-\rho)},$$

where $a := RS/\overline{X}$ and

$$\frac{b}{\overline{X}/S} = \begin{cases} 1 & \text{multiple-token} \\ \max[a, 1] & \text{single-token} \\ 1 + a & \text{single-message} \end{cases}.$$

In Fig. 9, we plot the normalized mean transfer time $T_f/(\overline{X}/S)$ against the throughput $\gamma = N\lambda\overline{X}/S$ of the system for the three operations; therefore $\rho = N\lambda b = \gamma \times b/(\overline{X}/S)$. Here we assume that

$$N = 50 \text{ stations} \quad ; \quad S = 4 \text{ megabits/sec} \quad ; \quad B = 1 \text{ bit latency per station.}$$

In the first case, we assume that $l = 1$ kilometer and $X = 1{,}000$ bits, which results in $\tau = 5$ microsec, $R = 17.5$ microsec, and $a = 0.07$. In the second case, we assume that $l = 10$ kilometers and $X = 100$ bits, which results in $\tau = 50$ microsec, $R = 62.5$ microsec, and $a = 2.5$. The difference in performance of the three operations is clearly shown in Fig. 9.

4. A Survey of Surveys

Until a few years ago I maintained a fairly complete (neither classified nor annotated) bibliography on polling models; it can be found at http://www.sk.tsukuba.ac.jp/~takagi/polling.html. (This does not mean that I have the hardcopy of all the publications in the list. So please do not order copies from me. But I welcome the correcting and updating of the information.) The bibliography contained over 700 publications, including journal and conference papers, books, theses, and technical reports. Since publications on polling continue to appear (although not at as high a pace as in the years around 1990), the number by now may total nearly 1,000.

When I began searching the literature on the analysis of polling models because I needed it for the performance evaluation of the token ring network in the early 1980s, I found that such models had been studied seemingly independently by researchers in different fields. I was fortunate in organizing all the results available at that time in a unified framework into a research monograph *Analysis of Polling Systems* published by the MIT Press in 1986 [17] and a survey paper "Queuing analysis of polling models" in the *ACM Computing Surveys* in 1988 [18]. Since then I have twice updated my survey to include subsequent developments: [19] covers the period until 1989 and [21] for 1990–1994. I also surveyed the applications to computer networks in [20].

Other researchers have also contributed surveys of polling models based on their own views. Let me mention some of them that have appeared after 1990. Grillo (1990) presents a classification of various polling models with an emphasis on communications [6]. Levy and Sidi (1990) summarize the analysis of polling models with variations, and describe the capabilities and limitations in their applications [12]. Rubin and Baker (1990) refer to both single and infinite buffer polling systems in their comprehensive review of media access control protocols for high-speed local area and metropolitan area communication networks [13]. Conti et al. (1993) show how single and infinite buffer polling systems with limited service rule have been used to model the Fiber Distributed Data Interface (FDDI) [4]. Boxma (1991) addresses static optimization problems with respect to server movement in polling systems [2]. Campbell (1991) demonstrates the difference between two "cyclical" queueing systems; one is the cycling server model (i.e., a polling system) and the other is the cycling customer model (i.e., a closed queueing network) [3]. Gupta and Günalay (1997) review the recent advances in the analysis of polling models in which the server uses system-state information to affect its behavior [5].

Polling models have been a part of many survey papers and books on the performance evaluation of communication networks. One such early survey is by Kobayashi and Konheim (1977) [10], and a more recent one is Kleinrock (1988) [9]. Relevant books are Akimaru and Kawashima [1], Hammond and O'Reilly [7], Hayes [8], Schwartz [14], and Stuck and Arthurs [15].

5. Future Directions

Let me conclude this essay with some personal views on possible future research directions. The analysis of polling models gained momentum as queueing systems that

are easy to understand, analyze, and extend. The study has been accelerated largely by applications to the modeling of communication, manufacturing and transportation systems. I believe that it is one of the few successful theoretical performance evaluation models developed in the last decades.

There still remain many unsolved mathematical problems in basic polling systems, such as stability conditions for multiple queues and the exact analysis of systems with limited service rules. Recent theoretical developments can be found in a special issue (Vol. 11, No. 1–2, 1992) of the *Queueing Systems* journal and in Volume 35 of *Annals of Operations Research* (1992). From the application viewpoint, one of the major directions of interest is the dynamic control and optimization of the server movement/action. Channel access protocols with polling schemes continue to appear; e.g., IEEE 802.12 100VG-Any LAN (demand priority) and IEEE 1394 arbitration. Approximate analysis of polling systems with multiple servers would also be of interest.

Acknowledgments

The author would like to thank Professor Robert B. Cooper of Florida Atlantic University, Dr. H. Richard Gail of IBM Thomas J. Watson Research Center, and Dr. Robert D. van der Mei of AT&T Labs for valuable comments on the draft of this article.

References

[1] Akimaru, H., and Kawashima, K. *Teletraffic: Theory and Applications*, Springer-Verlag, Heidelberg, 1993.

[2] Boxma, O. J., Analysis and optimization of polling systems. In: *Queueing, Performance and Control in ATM (ITC-13)*, J. W. Cohen and C. D. Pack (editors), pp.173–183, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1991.

[3] Campbell, G. M., Cyclic queueing systems. *European Journal of Operational Research*, Vol.51, No.2, pp.155–167, March 1991.

[4] Conti, M., Gregori, E., and Lenzini, L., Metropolitan area networks (MANs): protocols, modeling and performance evaluation. In: *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93*, L. Donatiello and R. Nelson (editors), pp.81–120, Lecture Notes in Computer Science 729, Springer-Verlag, Berlin, 1993.

[5] Gupta, D., and Günalay, Y., Recent advances in the analysis of polling systems. In: *Advances in Combinatorial Methods and Applications to Probability and Statistics*, N. Balakrishnan (editor), pp.339–360, Birkhäuser Boston, 1997.

[6] Grillo, D., Polling mechanism models in communication systems – some application examples. In: *Stochastic Analysis of Computer and Communication Systems*, H. Takagi (editor), pp.659–698, Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 1990.

[7] Hammond, J. L., and O'Reilly, P. J. P., *Performance Analysis of Local Computer Networks.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[8] Hayes, J. F., *Modeling and Analysis of Computer Communications Networks.* Plenum Press, New York, 1984.

[9] Kleinrock, L., Performance evaluation of distributed computer-communication systems. In: *Queueing Theory and Its Applications – Liber Amicorum for J. W. Cohen,* O. J. Boxma and R. Syski (editors), pp.1–57, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1988.

[10] Kobayashi, H., and Konheim, A. G., Queueing models for computer communications system analysis. *IEEE Transactions on Communications,* Vol.COM-25, No.1, pp.2–29, January 1977.

[11] Leibowitz, M. A., Queues. *Scientific American,* Vol.219, No.2, pp.96–103, August 1968.

[12] Levy, H., and Sidi, M., Polling systems: applications, modeling, and optimization. *IEEE Transactions on Communications,* Vol.38, No.10, pp.1750–1760, October 1990.

[13] Rubin, I., and Baker, J. E., Media access control for high-speed local area and metropolitan area communication networks. *Proceedings of the IEEE,* Vol.78, No.1, pp.168–203, January 1990.

[14] Schwartz, M., *Telecommunication Networks: Protocols, Modeling and Analysis.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.

[15] Stuck, B. W., and Arthurs, E., *A Computer and Communications Network Performance Analysis Primer.* Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[16] Sykes, J. S., Analysis of the communications aspects of an inquiry-response system. *AFIPS Conference Proceedings, 1969 Fall Joint Computer Conference,* Vol.35, pp.655–667, Las Vegas, Nevada, November 18–20, 1969.

[17] Takagi, H., *Analysis of Polling Systems.* The MIT Press, Cambridge, Massachusetts, 1986.

[18] Takagi, H., Queuing analysis of polling models. *ACM Computing Surveys,* Vol.20, No.1, pp.5–28, March 1988.

[19] Takagi, H., Queueing analysis of polling models: an update. In: *Stochastic Analysis of Computer and Communication Systems,* H. Takagi (editor), pp.267-318, Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 1990.

[20] Takagi, H., Application of polling models to computer networks. *Computer Networks and ISDN Systems,* Vol.22, No.3, pp.193–211, October 1991.

[21] Takagi, H., Queueing analysis of polling models: progress in 1990–1994. In: *Frontiers in Queueing: Models and Applications in Science and Technology,* J. H. Dshalalow (editor), pp.119–146 (Chapter 5), CRC Press, Boca Raton, Florida, 1997.